УДК 519.688 **DOI** 10.15622/sp.2019.18.3.557-581

А.А. Агафонов, В.В. Мясников

МЕТОД ОПРЕДЕЛЕНИЯ НАДЕЖНОГО КРАТЧАЙШЕГО ПУТИ В СТОХАСТИЧЕСКОЙ СЕТИ С ИСПОЛЬЗОВАНИЕМ ПАРАМЕТРИЧЕСКИ ЗАДАННЫХ УСТОЙЧИВЫХ РАСПРЕДЕЛЕНИЙ ВЕРОЯТНОСТЕЙ

Агафонов А.А., Мясников В.В. **Метод определения надежного кратчайшего пути в стохастической сети с использованием параметрически заданных устойчивых распределений вероятностей.**

Аннотация. Тенденция к увеличению количества транспортных средств, особенно в крупных городах, а также неспособность существующей дорожно-транспортной инфраструктуры распределять транспортные потоки, ведут к чрезмерной загрузке транспортных сетей и образованию дорожных заторов. Нерешенность данных проблем подчеркивает актуальность навигационных задач нахождения кратчайшего пути или оптимального маршрута движения. Несмотря на популярность этих задач, многие существующие коммерческие системы строят маршрут движения в детерминированных сетях, не учитывая зависящие от времени и стохастические свойства транспортных потоков. В работе рассматривается задача нахождения надежного маршрута движения в стохастической транспортной сети, максимизирующего вероятность прибытия в пункт назначения в течение заданного интервала времени. Надежный кратчайший путь учитывает дисперсию времени прохождения сегментов дорожной сети, что делает его более применимым для решения задач маршрутизации в транспортных сетях по сравнению со стандартными алгоритмами поиска кратчайшего пути, учитывающими только среднее время прохождения дорожных сегментов. Для описания времени прохождения сегментов дорожной сети предлагается использовать параметрически заданные устойчивые распределения вероятностей Леви. Использование устойчивых распределений позволяет перейти от операции вычисления свертки для определения надежности пути к пересчету параметров плотности распределения, что значительно сокращает время исполнения алгоритма. В работе решается задача нахождения адаптивного маршрута движения. Адаптивность подразумевает зависимость выбора следующего используемого дорожного сегмента от времени прибытия в вершину графа и определяется реальным состоянием дорожной сети. Экспериментальный анализ алгоритма, проведенный на крупномасштабной транспортной сети города Самара, показал, что представленный алгоритм позволяет значительно сократить время решения задачи нахождения надежного маршрута движения при незначительном увеличении времени проезда.

Ключевые слова: надежный кратчайший путь, стохастическая транспортная сеть, устойчивые распределения, распределение Леви

1. Введение. Навигационная задача нахождения кратчайшего пути в транспортной сети остается одной из наиболее актуальных задач в транспортных системах. Хотя существующие работы исследуют эту задачу в различных постановках, в том числе рассматривая зависящие от времени и стохастические транспортные сети, коммерческие системы работают с детерминированными сетями. В такой постановке путь, найденный между парой вершин отправления-назначения, считается оптимальным по определенным детерминированным критериям, например минимальное время движения или длина маршрута. Затраты на прохождение дорожных сегментов являются линейными и аддитивными, что позволяет использовать стандартные алгоритмы нахождения кратчайшего пути (например, Дейкстры [1]).

В реальной дорожно-транспортной сети, особенно в крупных мегаполисах, время (или скорость) прохождения дорожного сегмента должно рассматриваться как стохастическое, то есть движение отдельного транспортного средства зависит от многих факторов, в том числе от погодных условий, сезонности, времени дня, дорожных заторов, дорожно-транспортных происшествий и так далее. Если игнорировать стохастические свойства транспортных потоков, полученные результаты могут привести к выбору неоптимального маршрута движения и нежелательного позднего прибытия в пункт назначения. Однако учет не только ожидаемого (среднего) времени движения, но и дисперсии времени, то есть надежности маршрута, делает задачу оптимальной маршрутизации вычислительно сложной.

Задача нахождения надежного кратчайшего пути в зависящих от времени стохастических транспортных сетях рассматривалась в различных постановках [2-4], включая нахождение априорного оптимального пути или адаптивного маршрута движения. Однако существующие работы используют различные упрощения, в частности, не учитывают пространственно-временные корреляции времени прохождения дорожных сегментов. Другим недостатком существующих работ является проведение экспериментальных исследований на дорожных сетях малых размеров, так как предлагаемые решения не могут быть использованы в реальном времени из-за большой вычислительной сложности разработанных алгоритмов. Подробный обзор литературы, посвященной задаче нахождения кратчайшего пути в стохастической транспортной сети, представлен в следующем разделе.

В данной работе задача нахождения надежного кратчайшего пути рассматривается в следующей постановке: определить оптимальную стратегию навигации, максимизирующую вероятность прибытия в пункт назначения за выбранный бюджет поездки (в течение заранее определенного интервала времени). В работе предлагается использовать устойчивое распределение вероятностей для описания времени прохождения дорожных сегментов, что позволит заменить операцию вычисления свертки на пересчет параметров плотности распределения и значительно сократит время работы алгоритма.

Оставшаяся часть статьи построена следующим образом. Во втором разделе приведен краткий обзор литературы, посвященной теме исследования. Формальная постановка задачи и описание существующего алгоритма решения задачи представлены в третьем разделе. В четвертом разделе описан предложенный алгоритм нахождения надежного кратчайшего пути с использованием устойчивых распределений Леви. Описание и результаты экспериментальных исследований представлены в пятом разделе. Итоги и возможное направление дальнейших исследований приведены в заключении.

2. Обзор литературы. Алгоритмы маршрутизации в стохастических сетях предоставляют пользователям либо априорный оптимальный путь [2] или адаптивный маршрут движения [3]. В первом случае оптимальный путь определяется до момента отправления и не изменяется в процессе движения. При нахождении адаптивного маршрута путь может измениться при достижении очередного узла сети (перекрестка дорожной сети) после получения актуальной информации о времени прохождения сегментов дорожной сети.

В зависимости от используемого критерия оценки надежного кратчайшего пути используемые модели могут быть классифицированы следующим образом:

1. Модели с наименьшим ожидаемым временем движения (Least Expected Time — LET) [5-7]. Данные модели рассматривают ожидаемое время прохождения сегментов дорожной сети как критерий оценки для сравнения возможных путей.

2. Модели α-надежного пути. Целевая функция модели — минимизация интервала времени, необходимого для обеспечения прибытия в конечную вершину (пункт назначения) к выбранному моменту времени с заданной вероятностью α [2, 8, 9].

3. Модели с наибольшей надежностью движения [3, 10, 11]. Задача заключается в максимизации вероятности прибытия в конечную вершину (пункт назначения) в течение заранее определенного интервала времени (бюджета поездки). Часто эта проблема обозначается как SOTA (Stochastic On-Time Arrival).

Рассмотрим указанные модели подробнее. В [5] была рассмотрена задача нахождения пути с наименьшим ожидаемым временем движения в зависящих от времени стохастических сетях. В [7] была определена политика маршрутизации, определяющая, какой узел сети необходимо выбрать на следующем шаге с учетом актуального времени прохождения дорожных сегментов и текущего момента времени. Многокритериальный алгоритм A* для решения LET-задачи был предложен в [12]. Однако, зачастую необходимо не только минимизировать ожидаемое время в пути, но и учитывать потенциальный риск опоздания, так как путь с наименьшим ожидаемым временем движения может иметь высокую дисперсию.

Задача нахождения α-надежного кратчайшего пути была рассмотрена в [4]. Время прохождения дорожных сегментов задавались в виде коррелированных случайных величин с логнормальным законом распределения. Для нахождения надежного пути предложен гибридный генетический алгоритм на основе метода сопоставления моментов для аппроксимации параметров распределения времени прохождения пути. В статье [13] исследовалась проблема нахождения k-надежных кратчайших путей в стохастических сетях в условиях неопределенности времени прохождения. Задача k-надежных кратчайших путей расширяет классическую задачу k-кратчайших путей без циклов на стохастические сети, явно рассматривая надежность времени движения. Применялась концепция отклонения в пути для нахождения α-надежных кратчайших путей, для улучшения быстродействия использовался модифицированный алгоритм A*.

В статье [10] авторы сформулировали SOTA-проблему как задачу стохастического динамического программирования, для ее решения был применен стандартный метод последовательных аппроксимаций. Однако данный метод обладает плохой сходимостью. В статье [3] предложено точное решение SOTA-проблемы для сетей, в которых время прохождения сегментов является положительной величиной. Как и в работе [10], одним из этапов алгоритма является вычисление свертки, что является основной вычислительно сложной задачей. В общем виде свертка не может быть вычислена аналитически, и поэтому требуется схема дискретной аппроксимации. В [14] было предложено решение задачи нахождения априорного надежного пути с наибольшей надежностью движения, представлена адаптивная схема дискретизации для повышения эффективности свертки. Двухэтапный алгоритм решения задачи нахождения надежного пути предложен в [15]. На первом этапе оценивались верхние и нижние границы надежности времени движения, учитывая наименьшее ожидаемое время движения. На основе оцененного диапазона надежности было установлено эффективное правило доминирования и монотонное свойство целевой функции. На втором этапе использовался многокритериальный подход к установлению меток для определения наиболее надежного пути при различных сценариях принятия рисков. В [16] проблема смоделирована в виде Марковского процесса принятия решений. Предложен метод q-обучения, для определения функции полезности использовалась динамическая нейронная сеть. Заявлено, что метод может хорошо масштабироваться для крупномасштабных дорожных сетей. Задача нахождения надежного пути в расширенной постановке — минимизация функции риска перерасхода бюджета поездки — рассматривается в [17]. Представлены эффективные процедуры решения (на основе быстрого преобразования Фурье и алгоритма построения выпуклой оболочки) для вычисления приближенно-оптимальных стратегий навигаций. В работе [18] рассмотрена задача нахождения надежного пути в двух постановках. Формулируются эквивалентные задачи смешанного целочисленного линейного программирования, для их решения используется метод Лагранжевых релаксаций. Комбинированный метод Лагранжевых релаксаций для решения задачи маршрутизации предложен в [19]. В качестве критерия оптимизации рассматривается минимизация суммы среднего времени и среднеквадратического отклонения времени движения с учетом корреляции времени прохождения дорожных сегментов.

Повышения качества работы алгоритма навигации и снижения ошибки прогноза времени движения можно достичь, прогнозируя транспортные потоки на сегментах дорожной сети [20, 21]. В работе [22] представлена модификация решения [3], учитывающая актуальную и прогнозную информацию о транспортных потоках в сети.

Несколько исследований были посвящены ускорению работы алгоритма решения задачи SOTA. В статье [23] авторы представили несколько методов ускорения алгоритма решения SOTA-проблемы, включая усовершенствованные алгоритмы вычисления свертки с помощью быстрого преобразования Фурье и алгоритмы вычисления свертки с нулевой задержкой, а также методы определения оптимального порядка вычисления стратегии навигации. В [24] описана эвристика поиска адаптивного маршрута движения в стохастической сети. Представленный метод позволяет обеспечить наиболее вычислительно эффективную стратегию нахождения пути для общих распределений вероятностей. В [25] рассмотрены стохастические варианты двух методов предварительной обработки графа для решения задачи нахождения детерминированного кратчайшего пути, которые можно адаптировать к проблеме SOTA. Стратегия распараллеливания задачи с использованием графического процессора предложена в [26].

Несмотря на большое количество работ, посвященных данной проблеме, разработка эффективного алгоритма, который позволит быстро находить надежный кратчайший путь в крупномасштабной транспортной сети, остается предметом исследований. В данной работе представлен оригинальный подход к решению данной задачи, основанный на использовании параметрически заданных устойчивых распределений вероятностей Леви для описания времени прохождения сегментов дорожной сети.

3. Основные обозначения и постановка задачи. Мы рассматриваем зависящую от времени стохастическую улично-дорожную сеть в виде ориентированного графа $G = (N, A, P)$, где N — множество вершин графа, $|N|$ — количество вершин, A — множество ребер, $|A|$ — количество ребер, P — вероятностное описание времени прохождения ребер графа (т.е. сегментов транспортной сети).

В зависящих от времени стохастических сетях вес каждого сегмента $(i, j) \in A$ обычно представляется как случайная величина $T_{ij}(\tau)$ с зависящей от времени плотностью вероятности $p_{ij}^{\tau}(t)$.

Обозначим конечную вершину (пункт назначения) как $d \in N$, интервал времени, допустимый для достижения конечной вершины (т.е. бюджет поездки) как T. Оптимальная стратегия навигации определяется как стратегия максимизации вероятности прибытия в конечную вершину $d \in N$ из начальной (текущей) вершины $o \in N$ при наличии временного бюджета T.

Чтобы формально определить стратегию навигации, введем дополнительную величину $u_i(t)$ — вероятность прибытия в конечную вершину d из вершины i за время, не превышающее t [3, 10]. Тогда оптимальная стратегия навигации может быть сформулирована следующим образом:

$$\begin{aligned} u_i^{\tau}(t) = \max_{j \in N \wedge (i,j) \in A} \int\limits_0^t p_{ij}^{\tau}(\theta) u_j^{\tau+\theta}(t-\theta) d\theta, \\ \forall i \in N \setminus \{d\}, t \in [0, T], \tau \geqslant 0, \\ u_d^{\tau}(t) = 1, t \in [0, T], \tau \geqslant 0. \end{aligned} \tag{1}$$

Определим вероятность прохождения пути J за время t при условии, что поездка была начата в момент τ, как $u_J^{\tau}(t)$. Будем считать, что граф удовлетворяет условию стохастического FIFO, если выполняются следующие неравенства [3]:

$$\begin{aligned} & u_J^{t_1}(T) \geqslant u_J^{t_2}(T - (t_2 - t_1)), \\ & \forall J, \forall T, t_1, t_2 : 0 \leqslant t_1 \leqslant t_2, \, t_2 - t_1 \leqslant T. \end{aligned}$$

Согласно этому определению, незамедлительное начало движения по выбранному пути увеличивает вероятность прибытия в пункт назначения за выбранный бюджет поездки, по сравнению с отложенным началом движения.

Более формально модель поведения в графах, удовлетворяющих условию стохастического FIFO, характеризуется следующим положением: ожидание начала движения в нетерминальной вершине не удовлетворяет оптимальной стратегии навигации (1).

Для решения проблемы (1) в работе [3] был предложен дискретный алгоритм, который в виде псевдокода может быть записан следующим образом (алгоритм 1). На практике операции свертки вычисляются с использованием алгоритма быстрого преобразования Фурье.

Step 0. Initialization

$$k = 0$$

$$u_i^k(x) = 0, \quad \forall i \in N,\, i \neq d,\, x \in \mathbb{N},\, 0 \leqslant x \leqslant \frac{T}{\Delta t}$$

$$u_d^k(x) = 1, \quad x \in \mathbb{N},\, 0 \leqslant x \leqslant \frac{T}{\Delta t}$$

Step 1. Update

for $(\mathrm{k} = 1, 2, \ \ldots, \ L = \left[\frac{T}{\Delta t}\right])$ {

$$\tau^k = k\delta$$

$$u_d^k(x) = 1, \quad x \in \mathbb{N}, 0 \leqslant x \leqslant \frac{T}{\Delta t}$$

$$u_i^k(x) = u_i^{k-1}(x),$$

$$\forall i \in N, i \neq d, (i,j) \in A, x \in \mathbb{N}, 0 \leqslant x \leqslant \frac{\tau^k - \delta}{\Delta t}$$

$$u_i^k(x) = \max_j \sum_{h=0}^{x} p_{ij}(h) u_j^{k-1}(x-h)$$

$$\forall i \in N, i \neq d, (i,j) \in A, x \in \mathbb{N}, \frac{(\tau^k - \delta)}{\Delta t} + 1 \leqslant x \leqslant \frac{\tau^k}{\Delta t}$$

}

Алгоритм 1. Дискретный алгоритм решения SOTA

В алгоритме Δt — интервал дискретизации, δ — минимальное время прохождения дорожного сегмента в сети.

Тогда выбор следующей вершины j в графе дорожной сети (и, соответственно, следующего дорожного сегмента) с учетом оставшегося бюджета поездки t и вычисленного массива вероятностей прибытия $u_i(x)$ производится следующим образом:

$$j = \arg\max_{i \in N} u_i(t). \tag{2}$$

Наиболее вычислительно сложным этапом работы алгоритма является вычисление свертки $\sum_{h=0}^{x} p_{ij}(h)u_j^{k-1}(x-h)$.

В работе [22] в качестве плотности вероятности $p_{ij}(t)$ распределения времени движения на дорожном сегменте использовалось логнормальное распределение. В настоящей работе в качестве описания веса сегмента предлагается использовать устойчивое распределение Леви, что позволит заменить операцию свертки на пересчет параметров функции распределения.

4. Предлагаемый метод решения. В данном разделе представлен метод решения задачи определения надежного кратчайшего пути (1), максимизирующего вероятность прибытия в пункт назначения в течение заранее определенного интервала времени (бюджета поездки). Предлагаемое решение ограничивается рассмотрением не зависящих от времени стохастических сетей. В частности в следующих подразделах:

– представлено устойчивое распределение Леви для описания времени прохождения сегментов дорожной сети;

– показано, что операция свертки в алгоритме 1 может быть заменена пересчетом параметров функции распределения Леви;

– предложен алгоритм аппроксимации функции $\max_j(\cdots)$ в алгоритме 1 с помощью функции распределения Леви;

– представлен новый алгоритм решения задачи определения надежного кратчайшего пути;

– предложен способ ускорения работы алгоритма путем предварительного вычисления аппроксимаций.

4.1. Устойчивое распределение Леви. Распределение называется устойчивым, если линейная комбинация двух независимых случайных величин с этим распределением имеет то же распределение с точностью до коэффициента сдвига и масштаба.

Закон распределения случайной величины X называется устойчивым, если для любых идентично распределенных и независимых случайных величин X_1, X_2 и для любых действительных чисел $a > 0, b > 0$ найдутся числа $c > 0, b$ такие, что случайные величины $(aX_1 + bX_2)$ и $cX + d$ распределены одинаково.

В работе для задания веса каждого сегмента $(i, j) \in A$ используется устойчивое распределение Леви. Плотность вероятности распределения Леви для области определения $x \geqslant \mu$ имеет вид:

$$f(x; \mu, c) = \sqrt{\frac{c}{2\pi}} \; \frac{e^{-c/2(x-\mu)}}{(x-\mu)^{3/2}}, \tag{3}$$

где μ — коэффициент сдвига, c — коэффициент масштаба.

Функция распределения имеет вид:

$$F(x;\mu,c) = \operatorname{erfc}\left(\sqrt{c/\left(2(x-\mu)\right)}\right), \tag{4}$$

где $\operatorname{erfc}(z)$ — функция ошибок.

Пример плотности вероятности и функции распределения для распределения Леви с параметрами $\mu = 0, c = 1$ показан на рисунке 1.

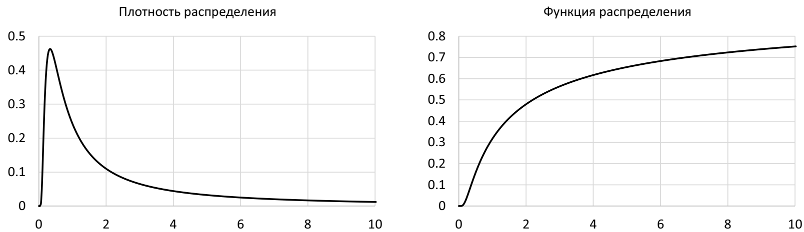

Рис. 1. Плотность вероятности и функция распределения для распределения Леви

Если $X_1 \sim \text{Levy}(\mu_1, c_1)$, $X_2 \sim \text{Levy}(\mu_2, c_2)$, то $X_1 + X_2 \sim \text{Levy}(\mu, c)$, где

$$\begin{aligned} \mu &= \mu_1 + \mu_2, \\ |c| &= \left(\sqrt{c_1} + \sqrt{c_2}\right)^2. \end{aligned} \tag{5}$$

Применительно к распределению времени прохождения сегментов дорожной сети коэффициент сдвига μ определяет минимальное время, за которое можно пройти сегмент, коэффициент масштаба определяет моду: $mode = c/3$.

4.2. Вычисление свертки. Покажем, что если время прохождения дорожных сегментов описывается устойчивым распределением Леви, операция свертки в алгоритме 1 может быть заменена на пересчет коэффициентов сдвига и масштаба распределения Леви.

Рассмотрим подробнее операцию свертки в алгоритме 1. Введем обозначение

$$u_{ij}(x) = \sum_{h=0}^{x} p_{ij}(h) u_j(x-h), \tag{6}$$

то есть $u_{ij}(x)$ — вероятность достижения конечной вершины d из вершины i за время x при движении по ребру (i, j).

Тогда выражение (1) можно записать в виде:

$$u_i(x) = \max_j \sum_{h=0}^{x} p_{ij}(h) u_j(x-h) = \max_j u_{ij}(x). \tag{7}$$

Рассмотрим сначала выражение $u_{ij}(x)$. Учитывая, что $u_d(x) = 1$, для ребер графа, входящих в конечную вершину d, мы можем получить:

$$u_{md}^k(x) = \sum_{h=0}^{x} p_{md}(h) u_d^{k-1}(x-h) = \sum_{h=0}^{x} p_{md}(h) = P_{md}(x), \quad \forall m \in N : \exists (m,d) \in A, \tag{8}$$

где $P_{md}(x)$ — функция распределения. То есть для вершин графа $m \in N$: $(m,d) \in A$, связанных с конечной вершиной d, вероятность достижения конечной вершины можно описать распределением Леви.

Далее, для предыдущих вершин графа $i : (i,m) \in A$ получим:

$$\begin{aligned} u_{im}(x) &= \sum_{h=0}^{x} p_{im}(h) u_m(x-h) = \sum_{h=0}^{x} p_{im}(h) \sum_{s=0}^{x-h} p_{md}(s) = \\ &= p_{im}(0) \sum_{s=0}^{x} p_{md}(s) + p_{im}(1) \sum_{s=0}^{x-1} p_{md}(s) + \cdots + p_{im}(x) p_{md}(0) = \\ &= \sum_{l=0}^{x} p_{im}(l) p_{md}(x-l) + \sum_{l=0}^{x-1} p_{im}(l) p_{md}(x-1-l) + \cdots \\ &+ \sum_{l=0}^{x-x} p_{im}(l) p_{md}(x-x-l) = \sum_{n=0}^{x} \sum_{l=0}^{x-n} p_{im}(l) p_{md}(x-n-l) = \\ &= \sum_{n=0}^{x} p_{im+md}(x-n) = P_{im+md}(x), \end{aligned}$$

где $p_{im+md}(t)$ — плотность вероятности суммы случайных величин.

Аналогично могут быть посчитаны значения $u_{ij}(x)$ $\forall i,j \in N$: $(i,j) \in A$, что позволяет заменить операцию свертки в алгоритме 1 на вычисление значения функции распределения. Значение коэффициентов масштаба и сдвига рассчитываются по формуле (5).

4.3. Аппроксимация функции максимума. Для определения надежного кратчайшего пути на следующем шаге необходимо получить оценку функции $u_i(x) = \max_j u_{ij}(x)$. Будем аппроксимировать значение $u_i(x)$ функцией распределения Леви. Обозначим аппроксимируемую функцию как $F^*(x)$, искомую функцию как $F(x;\hat{\mu},\hat{c})$. Тогда задача аппроксимации, в результате решения которой оцениваются искомые параметры $\hat{\mu}$ и $\hat{c}$, заключается в минимизации ошибки вида:

$$J = \epsilon^2 = \frac{1}{2}\sum_j \left(F^*(x_j) - F\left(x_j;\hat{\mu},\hat{c}\right)\right)^2 \to \min_{\hat{\mu},\hat{c}}. \qquad (9)$$

Примеры аппроксимации вероятности прибытия $u_i(x)$ представлены на рисунке 2. На рисунке 2 слева показан пример случая для двух возможных маршрутов движения в зависимости от времени прибытия в вершину, справа — для трех возможных маршрутов.

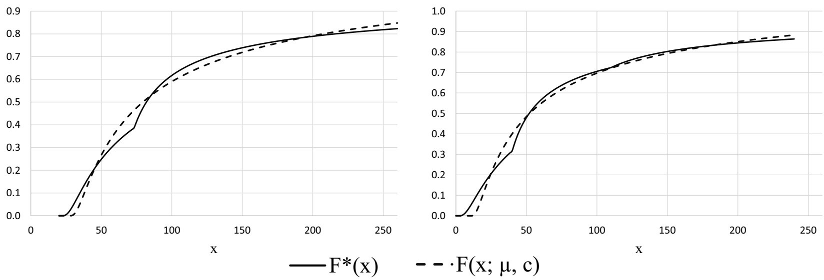

Рис. 2. Аппроксимация вероятности прибытия для случая двух и трех связанных вершин

Для минимизации ошибки (9) используется метод градиентного спуска. Для реализация градиентного спуска необходимо вычислить значения частных производных $\frac{\partial}{\partial\mu}F(x;\mu,c)$, $\frac{\partial}{\partial c}F(x;\mu,c)$.

Запишем частные производные в аналитическом виде.

$$\begin{aligned}
&\frac{\partial}{\partial\mu}F = \frac{\partial}{\partial\mu}\mathrm{erfc}\left(\sqrt{\frac{c}{2(x-\mu)}}\right) = -\frac{2\cdot c\cdot e^{\frac{c}{(2(x-\mu))}}}{\sqrt{\pi}\left(2m-2x\right)^2\sqrt{\frac{c}{2(x-\mu)}}},\\
&\frac{\partial}{\partial c}F = \frac{\partial}{\partial c}\mathrm{erfc}\left(\sqrt{\frac{c}{2(x-\mu)}}\right) = \frac{e^{\frac{c}{(2(x-\mu))}}}{\sqrt{\pi}\left(2m-2x\right)\sqrt{\frac{c}{2(x-\mu)}}}.
\end{aligned} \qquad (10)$$

Введем дополнительные обозначения для упрощения вида формул (10). Пусть

$$\beta(x,\mu,c) \triangleq \frac{c}{2\left(x-m\right)}, \quad \gamma(x,\mu,c) \triangleq \frac{\beta\cdot e^{-\beta}}{c\sqrt{\pi\beta}}. \qquad (11)$$

Тогда частные производные (10) примут вид:

$$\begin{aligned} \frac{\partial}{\partial \mu} F &= -\frac{2\beta^2 \cdot e^{-\beta}}{c\sqrt{\pi\beta}} = -2\beta\gamma, \\ \frac{\partial}{\partial c} F &= -\frac{\beta \cdot e^{-\beta}}{c\sqrt{\pi\beta}} = -\gamma. \end{aligned} \tag{12}$$

Таким образом, поиск искомой функции $F(x; \hat{\mu}, \hat{c})$ осуществляется путем минимизации ошибки (9) методом градиентного спуска с использованием соотношений (11), (12). Алгоритм градиентного спуска применительно к рассмотренной задаче (9) выглядит следующим образом (алгоритм 2).

Input data: $\theta_{init}, \mu_{init}, c_{init}, \varepsilon^{min}, I_0, I_1, M$;

$\theta = \theta_{init}, \mu = \mu_{init}, c = c_{init}$;
$\varepsilon = \sum\limits_j \left(F^*(x_j) - F\left(x_j; \mu, c\right)\right)^2$;
for $(i_0 = 1, 2, \cdots, I_0)$ {
  for $(i_1 = 1, 2, \cdots, I_1)$ {
    for $(j = 1, 2, \cdots, M)$ {
      $\Delta\mu = \Delta\mu - 2\left(F(x_j, \mu, c) - F^*(x_j)\right)\beta(x_j, \mu, c)\gamma(x_j, \mu, c)$;
      $\Delta c = \Delta c - \left(F(x_j, \mu, c) - F^*(x_j)\right)\gamma(x_j, \mu, c)$;
    }
    $\mu = \mu - \theta\Delta\mu$;
    $c = c - \theta\Delta c$;
  }
  $\varepsilon^{next} = \sum\limits_j \left(F^*(x_j) - F\left(x_j; \mu, c\right)\right)^2$;
  if $\varepsilon^{next} > \varepsilon$ {
    $\theta = \theta/2$;
  }
  if $\varepsilon^{next} < \varepsilon^{min}$ {
    break;
  }
  $\varepsilon = \varepsilon^{next}$;
}
$\hat{\mu} = \mu$;
$\hat{c} = c$;

Output data: $\hat{\mu}, \hat{c}$

Алгоритм 2. Метод градиентного спуска

В алгоритме 2 используются следующие параметры: θ_{init} — начальный шаг, μ_{init}, c_{init} — начальное приближение, ε^{min} — точность расчета, I_0, I_1 — число итераций, M — число используемых значений функции.

С учетом предложенного метода аппроксимации вероятность достижения конечной вершины $u_i(t)$ из вершины i также в итоге описывается функцией распределения Леви.

4.4. Алгоритм нахождения надежного пути. Для описания алгоритма нахождения надежного кратчайшего пути введем следующие обозначения.

Пусть время прохождения $p_{ij}(t)$ дорожного сегмента $(i, j) \in A, i \in N, j \in N$ описывается функцией распределения Леви с параметрами $P_{ij} = \left(\mu_{ij}^p, c_{ij}^p\right)$, вероятность достижения конечной вершины $u_{ij}(t)$ из вершины i за время t при движении по ребру (i, j) описывается функцией распределения Леви с параметрами $U_{ij} = \left(\mu_{ij}^u, c_{ij}^u\right)$, вероятность достижения конечной вершины $u_i(t)$ из вершины i за время t описывается функцией распределения Леви с параметрами $U_i = (\mu_i^u, c_i^u)$.

Для определения надежного кратчайшего пути, максимизирующего вероятность прибытия в пункт назначения в течение заранее определенного интервала времени (бюджета поездки), предлагается алгоритм навигации, состоящий из следующих шагов:

1. Для каждой помеченной вершины графа $i \in N$ выбираются все исходящие из нее ребра $(i, j) \in A$.

2. Считая известными параметры функции распределения U_j достижения конечной вершины d из вершины j и параметры функции распределения P_{ij} времени прохождения сегмента (i, j), рассчитываются параметры функции распределения U_{ij} достижения конечной вершины d из вершины i при движении по ребру (i, j) пересчетом коэффициентов масштаба и сдвига по формуле (5).

3. По рассчитанным параметрам функций распределения $U_{ij}, i \in N, j \in J, (i, j) \in A$ вычисляются параметры функции распределения U_i путем аппроксимации распределением Леви, как описано в подразделе 4.3.

4. Если рассчитанные параметры функции распределения U_i изменились, вершины $k \in N : (k, i) \in A$, связанные с вершиной i, помечаются для просмотра на следующей итерации алгоритма.

5. Если помеченные вершины отсутствуют — алгоритм завершает работу, иначе выполняется следующая итерация алгоритма (переход на шаг 1).

В виде псевдокода представленный алгоритм может быть записан следующим образом:

```
vertexes = {}, vertexesNextStep = {};
vertexes.add(d);
while (!vertexes.isEmpty()){
    for (vertex : vertexes) {
        params = {};
        for (edge : vertex.outgoingEdges()) {
            params.add(convolution(vertex.U, edge.P));
        }
        newU = approximate(params);
        if (vertex.U != newU) {
            vertex.U = newU;
            vertex.params = params;
            vertexesNextStep.addAll(incomingVertexes(vertex));
        }
    }
    vertexes = vertexesNextStep;
}
```

Алгоритм 3. Алгоритм решения SOTA с использованием распределений Леви

В алгоритме метод $convolution$ пересчитывает коэффициенты масштаба и сдвига (шаг 2 алгоритма), метод $approximate$ выполняет аппроксимацию функции max распределением Леви (шаг 3 алгоритма).

Следует отметить, что после завершения работы алгоритма каждая вершина i хранит список параметров функций распределения U_{ij}. Это необходимо для адаптивного выбора следующей вершины пути при движении по маршруту в зависимости от оставшегося бюджета поездки.

4.5. Предварительное вычисление аппроксимаций. Наиболее вычислительно сложным этапом работы алгоритма является аппроксимация распределением Леви на шаге 3. Учитывая, что распределение Леви имеет стандартную форму, обладающую следующим свойством:

$$f(x; \mu, c)dx = f(y; 0, 1)dy,$$

где y определяется как $y = \dfrac{x - \mu}{c}$, в работе предлагается провести предварительное вычислений аппроксимирующих функций Леви для различных значений коэффициентов сдвига и масштаба.

Пусть Δ — шаг, с которым изменялись коэффициенты сдвига в диапазоне $[0, 1]$ и коэффициенты масштаба в диапазоне $[\Delta, 1]$.

Выбор требуемых параметров аппроксимирующей функции состоит из следующих шагов:

– определение для вершины $i \in N$ набора параметров распределения Леви $\{U_{ij}\}$ для всех связанных вершин $j \in N : (i, j) \in A$ (шаг 3 алгоритма);

– определение минимального и максимального значения аргумента функции распределения:

$$\begin{aligned} x_{min} &= \min_j \mu^u_{ij}, \\ x_{max} &= \max_j \mu^u_{ij} + \alpha c^u_{ij}, \end{aligned} \tag{13}$$

где α — параметр, определяющий рассматриваемый диапазон функции распределения;

– расчет параметра масштабирования $scale = x_{max} - x_{min}$ и масштабирование параметров распределений Леви:

$$\hat{U}_{ij} = \left(\frac{\mu^u_{ij} - x_{min}}{scale}, \frac{c_{ij}}{scale} \right); \tag{14}$$

– выбор параметров ближайшей аппроксимирующей функции $\hat{U}^{approx} = (\hat{\mu}^{approx}, \hat{c}^{approx})$ по набору рассчитанных параметров $\hat{U}_{ij}$;

– обратное масштабирование:

$$U^{approx} = (\hat{\mu}^{approx} \cdot scale + x_{min}, \hat{c}^{approx} \cdot scale). \tag{15}$$

Результатом аппроксимации будут являться параметры U^{approx}.

5. Экспериментальные исследования. Целью экспериментальных исследований является сравнение результатов работы алгоритмов маршрутизации 1 (базовый алгоритм) и 3 (предложенный алгоритм), вычисляющих надежность пути через точное вычисление сверток и с помощью пересчета параметров функции Леви. Для сравнения алгоритмов необходимо оценить время работы процедуры маршрутизации построения надежного кратчайшего пути, а также оценить полученные маршруты движения.

На первом этапе исследований была проведена оценка ошибки аппроксимации целевой функции распределением Леви (рисунок 2) по критерию среднеквадратического отклонения:

$$\mathrm{RMSE} = \frac{1}{m} \sum_{i=1}^{m} \sqrt{\frac{1}{n} \sum_{j=1}^{n} \left(F_i^*(x_j) - F_i(x_j; \hat{\mu}_i, \hat{c}_i)\right)^2}, \tag{16}$$

где n — количество используемых отсчетов, m — количество аппроксимаций.

Ошибки аппроксимации считались отдельно для случаев, когда ребро графа (i, j) связано с двумя соседними ребрами (т.е. существует два возможных маршрута движения из вершины j, исключая движение в обратном направлении) и тремя соседними ребрами. Результаты приведены в таблице 1.

Таблица 1. Среднеквадратическое отклонение

	RMSE	Количество аппроксимаций
2 соседних ребра	0.0425	1694315
3 соседних ребра	0.0418	968040

Далее оценивались результаты работы алгоритмов маршрутизации.

Для проведения экспериментальных исследований разработанного метода использовалась крупномасштабная улично-дорожная сеть г. Самара, состоящая из 47274 дорожных сегментов и 18582 вершин. Часть улично-дорожной сети г. Самара показана на рисунке 3.

Рис. 3. Часть улично-дорожной сети г. Самара

Для определения параметров распределений времени прохождения сегментов дорожной сети использовались усредненные за десятиминутный интервал данные о скорости прохождения дорожных сегментов за два месяца. Для оценки прогнозного времени движения использовалось среднее время прохождения сегментов за два месяца, для оценки актуального времени прохождения использовались данные за конкретный день.

Для сравнения разработанного и базового алгоритмов по качеству решения задачи маршрутизации были выбраны 6 пар различных вершин отправления-прибытия на графе дорожной сети, после чего задача навигации решалась для каждой пары вершин и различных дней недели, времени начала движения и бюджета поездки. Вершины были выбраны таким образом, чтобы среднее время поездки составляло от 15 до 50 минут. Для каждого набора параметров задача решалась предлагаемым методом 3 и базовым алгоритмом 1. Всего было проведено 6300 экспериментов.

Гистограммы распределения разности времени движения, полученного путем решения задачи навигации предложенным алгоритмом и базовым алгоритмом на основе операции вычисления свертки, показаны на рисунке 4. Гистограммы приведены для прогнозного и актуального времени движения. Положительная разность соответствует проигрышу предложенного алгоритма (т.е. время движения по маршруту, рассчитанного предложенным алгоритмом, больше, чем время движения по маршруту, вычисленного базовым алгоритмом), отрицательная — выигрышу. В данном эксперименте рассматривались ситуации, когда время движения по маршрутам, найденным базовым и предложенным алгоритмами, не превышает бюджет поездки.

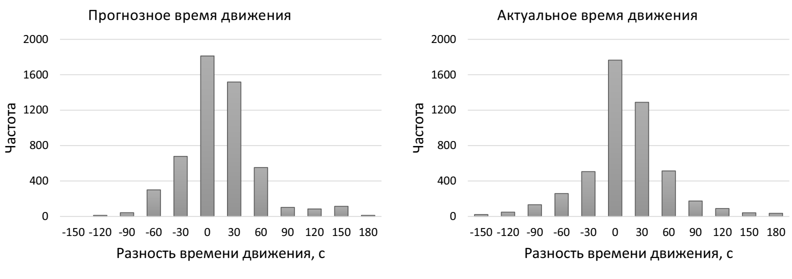

Рис. 4. Сравнение алгоритмов по фактическому времени движения транспортных средств

Как видно из представленных гистограмм, в большинстве случаев представленный алгоритм показывает тот же результат, что и алгоритм на основе вычисления свертки, либо предлагает маршрут движения с бóльшими временными затратами. Среднее время задержки на маршруте, найденном предложенным алгоритмом, приведено в таблице 2.

Далее оценивалось, насколько предложенный маршрут движения укладывался в требуемый бюджет поездки. Гистограмма распределения

Таблица 2. Среднее время задержки

	Прогнозное время движения	Актуальное время движения
Среднее время задержки, с	17.5	13.9

количества поездок в рамках временного бюджета (прибытие в указанный интервал) и вне его (прибытие вне интервала) показана в таблице 3 и на рисунке 5. Всего было проведено 6300 экспериментов. Из полученных результатов можно сделать вывод, что предложенный алгоритм ведет к нахождению маршрута движения вне выбранного бюджета чаще базового алгоритма примерно на 9%.

Таблица 3. Сравнение алгоритмов по затраченному временному ресурсу относительно бюджета поездки

	Пересчет параметров	Вычисление свертки
Вне интервала	0.154	0.064
В рамках интервала	0.846	0.936

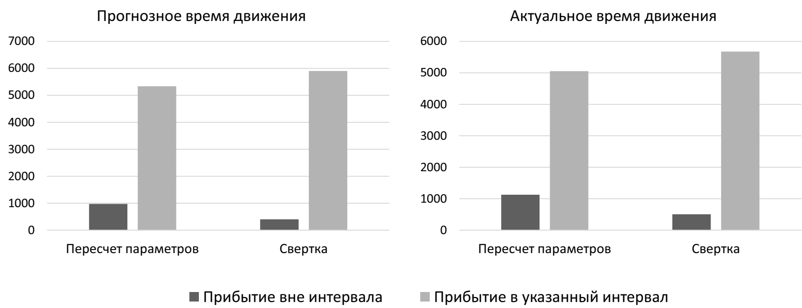

Рис. 5. Сравнение алгоритмов по затраченному временному ресурсу относительно бюджета поездки

На заключительном этапе экспериментального анализа проводилось измерение времени работы алгоритмов. Следует отметить, что время работы алгоритма-прототипа (Алгоритм 1) зависит от бюджета поездки T, который определяет количество итераций L. Число итераций разработанного алгоритма не зависит от заданного бюджета поездки. Среднее время работы алгоритмов представлено в таблице 4. Характеристики используемой ПЭВМ: процессор Intel Core i5-3740 3.20 GHz, оперативная память 16 ГБ, ОС – Windows 8.1.

Таблица 4. Сравнение времени работы алгоритмов

	Пересчет параметров	Вычисление свертки
Время работы, мс	606	23625

Подробнее среднее время работы базового алгоритма в зависимости от используемого бюджета поездки показано на рисунке 6.

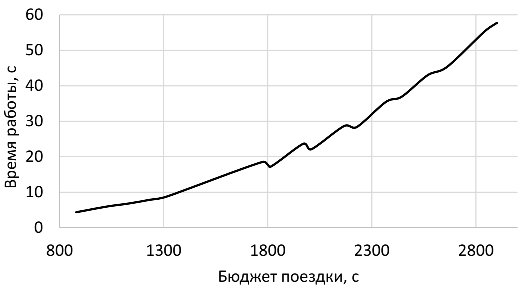

Рис. 6. Время работы базового алгоритма

Время работы разработанного алгоритма примерно в 40 раз меньше времени работы базового алгоритма и составляет в среднем 606 миллисекунд, что позволяет использовать алгоритм для решения задачи нахождения надежного кратчайшего пути в стохастической сети в режиме реального времени.

6. Заключение. В работе разработана модификация дискретного алгоритма решения задачи нахождения надежного маршрута движения в зависящей от времени стохастической транспортной сети. Для уменьшения времени работы алгоритма предложено использовать устойчивое распределение Леви для описания времени прохождения сегментов дорожной сети. Такой подход позволяет перейти от вычислительно сложной операции свертки к вычислению значения функции распределения для оценки вероятности прибытия в вершину назначения за указанное время.

Результаты экспериментального анализа на примере улично-дорожной сети г. Самара позволяют сделать вывод, что использование устойчивого распределения ведет к существенному (примерно в 40 раз) сокращению времени выполнения алгоритма при небольшом ухудшении результатов маршрутизации по критерию затраченного временного ресурса (в среднем время в пути увеличивается менее, чем на одну минуту). Среднее время работы алгоритма составляет 600 миллисекунд, что позволяет использовать его для решения задачи нахождения надежного маршрута движения, максимизирующего вероятность прибытия в указанный временной интервал для крупномасштабных транспортных сетей в режиме реального времени.

В качестве дальнейшего направления проведения исследований следует выделить модификацию алгоритма поиска надежного кратчайшего пути для использования в зависящей от времени стохастической сети, а также разработку параллельных алгоритмов решения задачи маршрутизации.

Литература

1. *Dijkstra E.W.* A note on two problems in connexion with graphs // Numerische Mathematik. 1959. vol. 1. no. 1. pp. 269–271.
2. *Nie Y.M., Wu X.* Shortest path problem considering on-time arrival probability // Transportation Research Part B: Methodological. 2009. vol. 43. no. 6. pp. 597–613.
3. *Samaranayake S., Blandin S., Bayen A.* A tractable class of algorithms for reliable routing in stochastic networks // Transportation Research Part C: Emerging Technologies. 2012. vol. 20. no. 1. pp. 199–217.
4. *Chen P., Tong R., Lu G., Wang Y.* The alpha-reliable path problem in stochastic road networks with link correlations: A moment-matching-based path finding algorithm // Expert Systems with Applications. 2018. vol. 110. pp. 20–32.
5. *Hall R.W.* The fastest path through a network with random time-dependent travel times // Transportation Science. 1986. vol. 20. no. 3. pp. 182–188.
6. *Fu L., Rilett L.* Fu L., Rilett L. Expected shortest paths in dynamic and stochastic traffic networks // Transportation Research Part B: Methodological. 1998. vol. 32. no. 7. pp. 499–516.
7. *Gao S., Chabini I.* Optimal routing policy problems in stochastic time-dependent networks // Transportation Research Part B: Methodological. 2006. vol. 40. no. 2. pp. 93–122.
8. *Chen A., Ji Z.* Path finding under uncertainty // Journal of Advanced Transportation. 2005. vol. 39. no. 1. pp. 19–37.
9. *Chen B.Y. et al.* Finding reliable shortest paths in road networks under uncertainty // Networks and Spatial Economics. 2013. vol. 13. no. 2. pp. 123–148.
10. *Fan Y., Nie Y.* Optimal routing for maximizing the travel time reliability // Networks and Spatial Economics. 2006. vol. 6. no. 3-4. pp. 333–344.
11. *Nie Y., Fan Y.* Arriving-on-time problem: Discrete algorithm that ensures convergence // Transportation Research Record. 2006. vol. 1964. no. 1. pp. 193–200.
12. *Chen P., Yin K., Sun J.* Application of finite mixture of regression model with varying mixing probabilities to estimation of urban arterial travel times // Transportation Research Record. 2014. vol. 2442. no. 1. pp. 96–105.
13. *Chen B.Y., Li Q., Lam W.H.* Finding the k reliable shortest paths under travel time uncertainty // Transportation Research Part B: Methodological. 2016. vol. 94. pp. 189–203.

14. *Nie Y.M., Wu X., Dillenburg J.F., Nelson P.C.* Reliable route guidance: A case study from chicago // Transportation Research Part A: Policy and Practice. 2012. vol. 46. no. 2. pp. 403–419.
15. *Chen B.Y. et al.* Most reliable path-finding algorithm for maximizing on-time arrival probability // Transportmetrica B: Transport Dynamics. 2017. vol. 5. no. 3. pp. 253–269.
16. *Cao Z. et al.* Maximizing the probability of arriving on time: A practical q-learning method // Thirty-First AAAI Conference on Artificial Intelligence. 2017. pp. 4481–4487.
17. *Flajolet A., Blandin S., Jaillet P.* Robust adaptive routing under uncertainty // Operations Research. 2018. vol. 66. no. 1. pp. 210–229.
18. *Yang L., Zhou X.* Optimizing on-time arrival probability and percentile travel time for elementary path finding in time-dependent transportation networks: Linear mixed integer programming reformulations // Transportation Research Part B: Methodological. 2017. vol. 96. pp. 68–91.
19. *Zhang Y., Khani A.* An algorithm for reliable shortest path problem with travel time correlations // Transportation Research Part B: Methodological. 2019. vol. 121. pp. 92–113.
20. *Agafonov A., Myasnikov V.* Traffic flow forecasting algorithm based on combination of adaptive elementary predictors // International Conference on Analysis of Images, Social Network and Texts. 2015. vol. 542. pp. 163–174.
21. *Агафонов А.А., Юмаганов А.С., Мясников В.В.* Анализ больших данных в геоинформационной задаче краткосрочного прогнозирования параметров транспортного потока на базе метода k ближайших соседей // Компьютерная оптика. 2018. Т. 42. № 6. С. 1101–1111.
22. *Агафонов А.А., Мясников В.В.* Метод определения надёжного кратчайшего пути в зависящей от вермени стохастической сети и его применение в геоинформационных задачах управления транспортом // Компьютерная оптика. 2016. Т. 40. № 2. С. 275–283.
23. *Samaranayake S., Blandin S., Bayen A.* Speedup techniques for the stochastic on-time arrival problem // OpenAccess Series in Informatic. 2012. vol. 25. pp. 83–96.
24. *Niknami M., Samaranayake S.* Tractable pathfinding for the stochastic on-time arrival problem // International Symposium on Experimental Algorithms. 2016. vol. 9685. pp. 231–245.
25. *Sabran G., Samaranayake S., Bayen A.* Precomputation techniques for the stochastic on-time arrival problem // 2014 Proceedings of the Sixteenth Workshop on Algorithm Engineering and Experiments (ALENEX). 2014. pp. 138–146.
26. *Abeydeera M., Samaranayake S.* GPU parallelization of the stochastic on-time arrival problem // 2014 21st International Conference on High Performance Computing (HiPC). 2014. pp. 1–8.

Агафонов Антон Александрович — канд. техн. наук, старший научный сотрудник, научно-исследовательская лаборатория геоинформатики и информационной безопасности, Самарский национальный исследовательский университет имени академика С.П. Королёва (Самарский уни-верситет); доцент, кафедра геоинформатики и информационной безопасности, Самарский национальный исследовательский университет имени академика С.П. Королёва (Самарский университет). Область научных интересов: геоинформаци-онные технологии, транспортное моделирование, веб-технологии. Число научных публикаций — 20. ant.agafonov@gmail.com; Московское шоссе, 34, 443086, Сама-ра, Российская Федерация; р.т.: +7(846)267-49-05.

Мясников Владислав Валерьевич — д-р физ.-мат. наук, профессор, кафедра геоинформатики и информационной без-опасности, Самарский национальный исследовательский университет имени академика С.П. Королёва (Самарский университет); ведущий научный сотрудник, ла-боратория математических методов обработки изображений, Институт систем обработки изображений РАН – филиал ФНИЦ «Кристаллография и фотоника» РАН. Область научных интересов: компьютерное зрение, распознавание образов и искусственный интеллект, искусственные нейронные сети, цифровая обработку сигналов и изображений, геоинформатика. Число научных публикаций — 200. vmyas@geosamara.ru; Московское шоссе, 34, 443086, Самара, Российская Федера-ция; р.т.: +7(846)267-49-05.

Поддержка исследований. Исследование выполнено при финансовой поддержке грантов РФФИ в рамках научных проектов № 18-07-00605 А, № 18-29-03135-мк в части постановки задачи и предлагаемого метода решения и Министерства науки и высшего образования РФ в рамках выполнения работ по Государственному заданию ФНИЦ «Кристаллография и фотоника» РАН (Соглашение № 007-ГЗ/Ч3363/26) в части «Экспериментальные исследования».

DOI 10.15622/sp.2019.18.3.557-581

A.A. AGAFONOV, V.V. MYASNIKOV

METHOD FOR RELIABLE SHORTEST PATH DETERMINATION IN STOCHASTIC NETWORKS USING PARAMETRICALLY DEFINED STABLE PROBABILITY DISTRIBUTIONS

Agafonov A.A., Myasnikov V.V. **Method for reliable shortest path determination in stochastic networks using parametrically defined stable probability distributions .**

Abstract. An increase in the number of vehicles, especially in large cities, and inability of the existing road infrastructure to distribute transport flows, leads to a higher congestion level in transport networks. This problem makes the solution to navigational problems more and more important. Despite the popularity of these tasks, many existing commercial systems find a route in deterministic networks, not taking into account the time-dependent and stochastic properties of traffic flows, i.e. travel time of road links is considered as constant. This paper addresses the reliable routing problem in stochastic networks using actual information of the traffic flow parameters. We consider the following optimality criterion: maximization of the probability of arriving on time at a destination given a departure time and a time budget. The reliable shortest path takes into account the variance of the travel time of the road network segments, which makes it more applicable for solving routing problems in transport networks compared to standard shortest path search algorithms that take into account only the average travel time of network segments. To describe the travel time of the road network segments, it is proposed to use parametrically defined stable Levy probability distributions. The use of stable distributions allows replacing the operation of calculating convolution to determine the reliability of the path to recalculating the parameters of the distributions density, which significantly reduces the computational time of the algorithm. The proposed method gives a solution in the form of a decision, i.e. the route proposed in the solution is not fixed in advance, but adaptively changes depending on changes in the real state of the network. An experimental analysis of the algorithm carried out on a large-scale transport network of Samara, Russia, showed that the presented algorithm can significantly reduce the computational time of the reliable shortest path algorithm with a slight increase in travel time.

Keywords: Reliable Shortest Path, Stochastic Transportation Network, Stable Distributions, Levy Distribution.

Agafonov Anton Aleksandrovich — Ph.D., Senior Resercher, Science Research Laboratory of Geoinformatics and Infor-mation Security, Samara University; Associate Professor, Department of Geoinformat-ics and Information Security, Samara University. Research interests: geoinformatics, transport modelling and web-technologies. The number of publications — 20. ant.agafonov@gmail.com; 34, Moskovskoye Shosse, 443086, Samara, Russian Federa-tion; office phone: +7(846)267-49-05.

Myasnikov Vladislav Valerievich — Ph.D., Dr.Sci., Professor, Department of Geoinformatics and Information Security, Samara National Research University; Leading Researcher, Laboratory of Mathematical Methods for Image Processing, Institute of Image Processing Systems of the Russian Academy of Sciences – a branch of the Russian Academy of Sciences “Crystallography and Photonics”. Research interests: omputer vision, pattern recognition and artificial intelligence, artificial neural networks, digital processing of signals and images, and geoinformatics. The number of publications — 200. vmyas@geosamara.ru; 34, Mos-kovskoye Shosse, 443086, Samara, Russian Federation; office phone: +7(846)267-49-05.

Acknowledgements. The work was funded by the Russian Foundation for Basic Research under research projects Nos. 18-07-00605, 18-29-03135 ("Basic theoretical propositions and the proposed method") and the RF Ministry of Science and Higher Education within the State assignment to the FSRC «Crystallography and Photonics» RAS under agreement 007-G3/Ch3363/26 ("Experiment analysis").

References

1. Dijkstra E.W. A note on two problems in connexion with graphs. *Numerische Mathematik*. 1959. vol. 1. no. 1. pp. 269–271.
2. Nie Y.M., Wu X. Shortest path problem considering on-time arrival probability. *Transportation Research Part B: Methodological*. 2009. vol. 43. no. 6. pp. 597–613.
3. Samaranayake S., Blandin S., Bayen A. A tractable class of algorithms for reliable routing in stochastic networks. *Transportation Research Part C: Emerging Technologies*. 2012. vol. 20. no. 1. pp. 199–217.
4. Chen P., Tong R., Lu G., Wang Y. The alpha-reliable path problem in stochastic road networks with link correlations: A moment-matching-based path finding algorithm. *Expert Systems with Applications*. 2018. vol. 110. pp. 20–32.
5. Hall R.W. The fastest path through a network with random time-dependent travel times. *Transportation Science*. 1986. vol. 20. no. 3. pp. 182–188.
6. Fu L., Rilett L. Expected shortest paths in dynamic and stochastic traffic networks. *Transportation Research Part B: Methodological*. 1998. vol. 32. no. 7. pp. 499–516.
7. Gao S., Chabini I. Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research Part B: Methodological*. 2006. vol. 40. no. 2. pp. 93–122.
8. Chen A., Ji Z. Path finding under uncertainty. *Journal of Advanced Transportation*. 2005. vol. 39. no. 1. pp. 19–37.
9. Chen B.Y. et al. Finding reliable shortest paths in road networks under uncertainty. *Networks and Spatial Economics*. 2013. vol. 13. no. 2. pp. 123–148.
10. Fan Y., Nie Y. Optimal routing for maximizing the travel time reliability. *Networks and Spatial Economics*. 2006. vol. 6. no. 3-4. pp. 333–344.
11. Nie Y., Fan Y. Arriving-on-time problem: Discrete algorithm that ensures convergence. *Transportation Research Record*. 2006. vol. 1964. no. 1. pp. 193–200.
12. Chen P., Yin K., Sun J. Application of finite mixture of regression model with varying mixing probabilities to estimation of urban arterial travel times. *Transportation Research Record*. 2014. vol. 2442. no. 1. pp. 96–105.
13. Chen B.Y., Li Q., Lam W.H. Finding the k reliable shortest paths under travel time uncertainty. *Transportation Research Part B: Methodological*. 2016. vol. 94. pp. 189–203.
14. Nie Y.M., Wu X., Dillenburg J.F., Nelson P.C. Reliable route guidance: A case study from Chicago. *Transportation Research Part A: Policy and Practice*. 2012. vol. 46. no. 2. pp. 403–419.
15. Chen B.Y. et al. Most reliable path-finding algorithm for maximizing on-time arrival probability. *Transportmetrica B: Transport Dynamics*. 2017. vol. 5. no. 3. pp. 253–269.
16. Cao Z. et al. Maximizing the probability of arriving on time: A practical q-learning method. Thirty-First AAAI Conference on Artificial Intelligence. 2017. pp. 4481–4487.
17. Flajolet A., Blandin S., Jaillet P. Robust adaptive routing under uncertainty. *Operations Research*. 2018. vol. 66. no. 1. pp. 210–229.
18. Yang L., Zhou X. Optimizing on-time arrival probability and percentile travel time for elementary path finding in time-dependent transportation networks: Linear mixed integer programming reformulations. *Transportation Research Part B: Methodological*. 2017. vol. 96. pp. 68–91.

19. Zhang Y., Khani A. An algorithm for reliable shortest path problem with travel time correlations. *Transportation Research Part B: Methodological*. 2019. vol. 121. pp. 92–113.
20. Agafonov A., Myasnikov V. Traffic flow forecasting algorithm based on combination of adaptive elementary predictors. International Conference on Analysis of Images, Social Network and Texts. 2015. vol. 542. pp. 163–174.
21. Agafonov A.A., Yumaganov A., Myasnikov V.V. [Big data analysis in a geoinformatic problem of short-term traffic flow forecasting based on a k nearest neighbors method]. *Komp'yuternaya optika – Computer optics*. 2018. Issue 42. vol. 6. pp. 1101–1111. (In Russ.).
22. Agafonov A.A., Myasnikov V.V. [Method for the reliable shortest path search in time-dependent shochastic network and its application to gis-based traffic control]. *Komp'yuternaya optika – Computer optics*. 2016. Issue 40. vol. 2. pp. 275–283. (In Russ.).
23. Samaranayake S., Blandin S., Bayen A. Speedup techniques for the stochastic on-time arrival problem. *OpenAccess Series in Informatic*. 2012. vol. 25. pp. 83–96.
24. Niknami M., Samaranayake S. Tractable pathfinding for the stochastic on-time arrival problem. International Symposium on Experimental Algorithms. 2016. vol. 9685. pp. 231–245.
25. Sabran G., Samaranayake S., Bayen A. Precomputation techniques for the stochastic on-time arrival problem. 2014 Proceedings of the Sixteenth Workshop on Algorithm Engineering and Experiments (ALENEX). 2014. pp. 138–146.
26. Abeydeera M., Samaranayake S. GPU parallelization of the stochastic on-time arrival problem. 2014 21st International Conference on High Performance Computing (HiPC). 2014. pp. 1–8.